

The General Linear Model - ANOVA

Part 1 (Video 3)

Dr Andrew J. Stewart

E: drandrewjstewart@gmail.com

T: [@ajstewart_lang](#)

G: [ajstewartlang](#)



Repeated Measures ANOVA

- Let's imagine we have an experiment where we asked 32 participants to learn how to pronounce words of differing levels of complexity - Very Easy, Easy, Hard, and Very Hard.
- They were presented with these words in an initial exposure phase. After a 30 minute break we tested participants by asking them to say the words out loud when they appeared on a computer screen.
- We want to know whether there is a difference in their response times as a function of each level of word complexity.

Repeated Measures ANOVA

```
rm_data <-  
read_csv("https://raw.githubusercontent.com/ajstewartlang/11_glm_anova_pt1/master/data/rm_data.csv")  
head(rm_data)
```

```
# A tibble: 6 x 3
```

	Participant	Condition	RT
	<dbl>	<chr>	<dbl>
1	1	Very Easy	1.25
2	2	Very Easy	1.16
3	3	Very Easy	1.12
4	4	Very Easy	1.33
5	5	Very Easy	1.16
6	6	Very Easy	1.15

Repeated Measures ANOVA

```
rm_data_tidied <- rm_data %>%  
  mutate(Condition = factor(Condition))  
head(rm_data_tidied)
```

```
# A tibble: 6 x 3
```

	Participant	Condition	RT
	<dbl>	<fct>	<dbl>
1	1	Very Easy	1.25
2	2	Very Easy	1.16
3	3	Very Easy	1.12
4	4	Very Easy	1.33
5	5	Very Easy	1.16
6	6	Very Easy	1.15

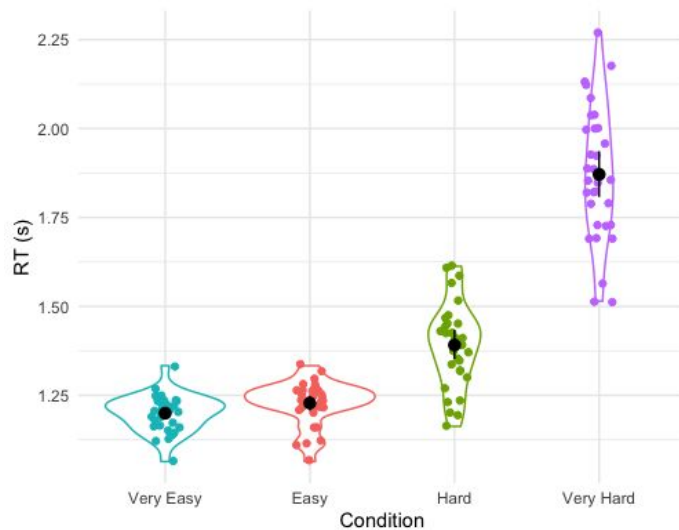
Summarising our Data

```
rm_data_tidied %>%  
  group_by(Condition) %>%  
  summarise(mean = mean(RT), sd = sd (RT))
```

```
# A tibble: 4 x 3  
  Condition  mean    sd  
  <fct>      <dbl> <dbl>  
1 Easy       1.23 0.0610  
2 Hard       1.39 0.118  
3 Very Easy  1.20 0.0511  
4 Very Hard  1.87 0.187
```

Visualising our Data

```
rm_data_tidied %>%  
  ggplot(aes(x = fct_reorder(Condition, RT), y = RT, colour = Condition)) +  
  geom_violin() +  
  geom_jitter(width = .1) +  
  guides(colour = FALSE) +  
  stat_summary(fun.data = "mean_cl_boot", colour = "black") +  
  theme(text = element_text(size = 13)) +  
  theme_minimal() +  
  labs(x = "Condition", y = "RT (s)")
```



Modelling our Data

```
> rm_model <- aov_4(RT ~ Condition + (1 + Condition | Participant), data = rm_data_tidied)
```

```
> summary(rm_model)
```

```
Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
```

	Sum Sq	num Df	Error Df	SS den	F value	Pr(>F)
(Intercept)	259.07	1	0.50313	31	15962.33	< 2.2e-16 ***
Condition	9.27	3	1.20624	93	238.23	< 2.2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Mauchly Tests for Sphericity
```

	Test statistic	p-value
Condition	0.38404	3.0211e-05

```
Greenhouse-Geisser and Huynh-Feldt Corrections
```

```
for Departure from Sphericity
```

	GG eps	Pr(>F[GG])
Condition	0.65596	< 2.2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	HF eps	Pr(>F[HF])
Condition	0.7000534	3.359493e-31

Our Effect Size Measure

```
> anova(rm_model)
Anova Table (Type 3 tests)

Response: RT
          num Df den Df      MSE      F      ges      Pr(>F)
Condition 1.9679 61.004 0.019773 238.23 0.84431 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The effect size is measured by ges and is the recommended effect size measure for repeated measures designs (Bakeman, 2005). Note the dfs in this output are always corrected as if there is a violation of sphericity (violated when the variances of the differences between all possible pairs of within-subject conditions (i.e., levels of the independent variable) are **not** equal) - to be conservative (and to avoid Type I errors) we might be better off to always choose these corrected dfs.

Interpreting our Model

To determine what's driving the effect we can use `emmeans::emmeans()` to run pairwise comparisons (note, default is Tukey correction and we have explicitly asked for Bonferroni below).

```
> emmeans(rm_model, pairwise ~ Condition, adjust = "Bonferroni")
```

```
$emmeans
  Condition emmean      SE  df lower.CL upper.CL
Easy       1.23 0.0208 123    1.18    1.28
Hard       1.39 0.0208 123    1.34    1.44
Very.Easy  1.20 0.0208 123    1.15    1.25
Very.Hard  1.87 0.0208 123    1.82    1.92
```

```
Warning: EMMs are biased unless design is perfectly balanced
```

```
Confidence level used: 0.95
```

```
Conf-level adjustment: bonferroni method for 4 estimates
```

```
$contrasts
```

```
  contrast      estimate      SE  df t.ratio p.value
Easy - Hard      -0.1633 0.0285  93  -5.735 <.0001
Easy - Very.Easy  0.0285 0.0285  93   1.000 1.0000
Easy - Very.Hard  -0.6430 0.0285  93 -22.584 <.0001
Hard - Very.Easy  0.1917 0.0285  93   6.734 <.0001
Hard - Very.Hard  -0.4797 0.0285  93 -16.849 <.0001
Very.Easy - Very.Hard -0.6715 0.0285  93 -23.584 <.0001
```

```
P value adjustment: bonferroni method for 6 tests
```